Introducing Structured Information Handling in Automotive EE Development

Diana Malvius, Ola Redell and Sofia Ritzén
{malvius, ola, sofia}@md.kth.se
Department of Machine Design
KTH, 100 44 Stockholm

Abstract. One way for the automotive industry to cope with the demand of a more structured information handling is to adjust model-based development (MBD) to multidisciplinary needs. Many of the issues faced in this transition are as much organizational and managerial as they are technical. In a case study carried out at a global automotive manufacturer a project to improve the electrical and electronics (EE) development has been followed and analyzed. The project originated from different needs identified by management in their ongoing work effort towards MBD as well as by developers who experienced that tools did not support their work situation. This paper describes how the introduction of a new tool support was made in a project carried out within EE development, further it reports on benefits achieved by using the tool. Both the effect on the work of affected EE developers and the expansion of a new information model are discussed, leaving important implications for management. Success keys for putting new support tools into practice are identified here and include; a bottom-up approach, user involvement from the beginning, focus on the individual needs and adaptation to current work practice. Further, management support and adequate resources are essential for extracting long-term benefits.

Introduction

Even though cars have incorporated large numbers of electrical features for many years, the automotive industry still has not been able to fully adapt its development process to the specific needs of electrical and electronics (EE) and software development. Due to an increased product complexity new demands are placed on the product development process. The needs among developers for an altered development process originate from the EE discipline in which software development is the major challenge. The growing needs of information handling within the EE development is one important factor that calls for an updated development process and enhanced tool support. While information handling of products with high mechanical content has strong support in mature technologies such as Product Data Management (PDM) and Product Lifecycle Management (PLM) systems, the correspondence in EE development is weak. This is a well-known problem that is currently addressed by both industry and academia in a number of international research projects. These projects cover for example the development of open standards for information models and platforms for EE architectures (Autosar 2005; EAST-EEA 2005). The projects promote model-based development (MBD) and aim at making the development of advanced software based functionality in vehicles more flexible by e.g. enabling the use of COTS (commercial off-the-shelf) components and simplified replacement of subcontractors.

Changes in technology content in products and changes in development processes follow each other, however, in most firms without little consciousness on how changes in the organizing of
activities and people are conducted (Sarv 1991). Different change strategies must be utilized depending on change category and expected results from changes (Porras and Robertson 1992; Norrgren and Rendahl 1995). Theories on such strategies as well as criteria for successful change procedures (Kotter 1996) can assist in evaluating performed changes. The introduction of tools characterizes a well defined change aiming for continuous improvements which can be specifically evaluated with implementation guidelines (Beskow and Ritzén 2000).

The work presented in this paper describes a study carried out within the EE development at an automotive manufacturer belonging to a larger global group. The purpose with this paper is to describe and discuss a small scale introduction of a new tool support for EE development and to analyze possible implications of this change on work conditions for concerned developers. The objective is to identify strengths and weaknesses of the change work preparing for a possible large scale implementation. Another objective is to illustrate the benefit of introducing a model-based information handling approach in EE development.

Model-based development. MBD, which is currently being adopted by the automotive industry, is one way to approach the increasing problem with information handling. Efforts to work more model-based have come far and the introduction of MBD has been simplified by its apparent advantages including possibilities to reuse models and manage complexity (Törngren and Larses 2005). However, as mentioned by Knodel et al. (2005), before being introduced into industrial practice, if developers are really going to use MBD in their work it first needs to be tailored to current processes. Furthermore, to gain development efficiency, it is important to first achieve a good understanding of how to apply MBD and what the common pitfalls are (Törngren and Larses 2005).

Moving from a document oriented view where text documents are the main source of information, to a model-based view where information consists of different modelling artefacts, is not trivial. A disciplined approach is required in order to identify and structure the required model information. Information for different disciplines is usually structured in different ways and different systems require different types of models. Further, much of the model data that is handled for different systems is interrelated, or even identical, which leads to a need to approach data consistency and data sharing issues. For these reasons, to succeed with the introduction of MBD, it is very important to agree upon a well defined EE information model that covers the needs of all concerned stakeholders.

It is necessary for an organization to think through its information model and to take control of the information in order to be successful in its information handling (Davenport and Prusak 1988; Stevens et al. 1998). When product information is not sufficiently structured, system designers tend to spend much focus on the documents themselves looking for the correct information. This problem grows when the amount of information increases. Instead, a structured information handling should allow developers to focus on information and should assist them in, for example, tracing modelling information between different disciplines.

Contextual settings

When studying complex product development it is important to consider organization, product and process (Eppinger and Salminen 2001). These domains are more closely discussed in the following sections, with a specific perspective of automotive EE development.
**Organization.** Complex product development settings like in the automotive industry put new demands on organizational structure and roles. Due to different technology understanding and incoherent use of terminology, information sharing between individuals is made more difficult when multiple engineering disciplines need to coordinate their work. For a multi-brand company where several product as well as system platforms have to be considered, problems with multidisciplinary work are made worse by the extended need of information transfer across organizations.

In the studied company the R&D unit is organized according to a product system structure where the car is divided into different technical sub systems e.g. Infotainment, Powertrain and Chassis as is illustrated in Figure 1. People involved in the development of a particular system, regardless of platform belonging, are geographically located together. The purpose of this is to simplify carry over by increasing the information transfer between platforms and establishing a common work procedure for the entire platform development.

**Product.** Vehicular systems are inherently complex with sub-systems spanning over a multitude of engineering disciplines such as mechanical, hydraulic, combustion, electronic and software engineering. However, in vehicle manufacturing today, electronics and software constitute the single biggest expense and account for one third of a vehicle’s total value. Even though already large, this number is expected to grow continuously in the near future (George and Wang 2002). A consequence of the large number of EE-based functions has been an immense increase in the amount of product data that needs to be handled during EE development.

**Process.** Today’s development processes originate in long traditions of development of mechanical products and are not sufficiently adapted to the needs of EE development. When software becomes a crucial part of mechanical products, software engineering also needs to be well integrated into the product development and to succeed with this integration, new management approaches are required (Rauscher and Smith 1995; Stevens et al. 1998; Knippel and Schulz 2004). Work procedures include processes such as tasks and activities, formal roles and defined responsibilities and accessible tools. It is important to realize that there is a need to
adjust these routines in order to meet the demands of complex product development. The generic procedure for complex system development according to a V-model (VDI-richtlinien 2003) consists of three main phases; concept phase that consists of function development, detailed design phase and system integration phase. At the studied company an adapted version of the V-model is used as formal development process. System design and verification is done iteratively and requirements are released and reviewed in batches (Figure 2).

![Figure 2. The V-model adopted at the studied company.](image)

**Research approach.** The following study, conducted in 2005, is based on semi-structured interviews. Each interview covered a number of project related themes prepared with specific questions around which the interviewees were allowed to initiate further discussions. Every person involved in the change project including EE developers, project leaders and external company consultants have taken part in the interview study. The interviews were conducted both at the studied company and at the system tool supplier company. The interviews were carried out by the authors and lasted on average for 90 minutes. Verification of the results has been made through active feedback from interviewees and the company management within EE development.

**The change project – Model-Based EE Development**

In the following the details of the studied change project Model-Based EE Development, MBEED, will be described and discussed.

**Experienced needs among EE developers.** Due to difficulties in their daily work EE developers in the studied organization experienced an urgent need for a new tool that could support their work procedures. One limiting factor has been the use of word processors to handle all requirement specifications, something that is not sufficient to meet the developers’ needs when system requirement documents exceed 1000 pages in size, filled with cross references. With information spread in several documents, it is not only cumbersome and time consuming to find correct information, but also difficult to keep the information up to date and consistent at all
times. Figure 3 shows an illustration of what a simplified document tree currently looks like at the studied company. If the need of an advanced information handling could be met, problems with non-traceable and inconsistent information would be solved or at least facilitated.

Another difficulty concerning the studied development process was the company’s use of UML diagrams for high level descriptions of objects for which specifications were to be produced. Not only was UML difficult to learn for the average engineer, but the UML language also uses a terminology very different from the one automotive engineers use in their daily work and the developers were not willing to adapt their terminology to an external system standard. Therefore, they felt reluctant to use UML. Another problem was a limited availability of the UML tool – in reality only a few UML modellers took care of all modelling – leading to the overall consequence that there was no strong enough incitement for the developers to strictly follow the dictated UML-based process. The last circumstance was a major source for inconsistencies in the UML diagrams as well as in the written specifications.

**Objectives with the tool introduction.** When the new tool support was introduced, one objective was to improve the system requirements process through a structured handling of information in smaller entities than earlier. This would allow release and subsequent review of individual requirements, in the pace that refined requirements were prepared, instead of batch-wise releases of huge requirement documents. Further, the new support tool was to be adaptable to the existing development process and support regular work without complications. From middle management, another objective was to point out similarities between different organizational units in the informal work procedures. This was expected to strengthen the company objective of establishing one prevailing development process that could easier be
further refined later. This would in turn lead to better conditions for handling increased product complexity.

**Project design and choice of key users.** The change work originated from a small number of EE developers who drew attention to their poor work situation and eventually persuaded management of the need to take on strong measures. This resulted in the initiation of a project with the purpose to introduce a new tool for information handling in EE development. It was decided that the tool would initially be tested within a limited group of developers closely connected to EE development (in the following referred to as key users). There were several reasons for this; for one it was considered important that the introduction would be based on the developers’ own will and interest in change, and not a simple enforcement of a new tool usage within an entire department; second the project was not allowed to negatively inflict on the ongoing work, demanding the possibility for key users to work with parallel tools during the introduction phase.

The project was designed to cover representative parts of the V-model with the objective to get a good perception of how the organization’s information model for EE development ought to be designed. Key users were therefore chosen from different organizational units (Figure 1) and different levels in the V-model (Figure 2); function, system and component levels were covered in the project. One key user worked at the function specification level in the Powertrain unit and needed an information handling tool to structure and collect data for an early dependability analysis. Starting with non functional requirements on dependability, functional requirements were produced through analysis. For this key user, traceability was a main concern in order to clarify the origins of functional requirements that were otherwise difficult to understand. Another key user, a system-level developer within the Infotainment unit, used the tool to specify and release requirements to a key user with component responsibilities (see Figure 2 for V-model reference). In this case, the objects handled by the two key users were partial requirements in textual format. The system designer released requirements that the component designer received and either accepted or rejected. Accepted requirements were directly attached to the component requirements managed in the tool that the component designer was responsible of preparing. The system designer worked with more than one component designer and therefore had to prepare a traditional written system specification as well as providing requirements in the new tool. This was not experienced as a problem. The component designer included all requirements that he received from all system designers he worked with, in the new information handling tool, and when all requirements were accepted he used the tool to automatically generate a written requirement specification in PDF-format that was delivered to a sub-contractor implementing the component.

One important fact that contributed to what the respondents experienced as a positive outcome of the project was that a personal need for a new tool had to be established first in order for an EE developer to take part in the project. This motivated chosen key users to take an active part in the change work throughout the entire project. An articulated project purpose, uniform for both management and project members, resulted in everyone striving towards the same objective. Further, the fact that a representative choice among EE developers was successfully made led to end results in terms of partial information models and tailored tool clients that a broad group of EE developers working under similar circumstances will be able to benefit from. However, the parallel development of multiple information models covering different parts of the modeling needs of EE development was identified as a potential problem. The reason for this was that
different islands of users with their related information models would evolve and in the end show to be incompatible and difficult to integrate into a complete information model. Further, the fact that someone takes an interest in one’s work helps to make individuals visible in the organization and contributes to inspire and motivate people to do a better job. One spin-off effect due to this confirmation act was improved communication between departments. People started talking cross disciplines and comparing experiences, trying to cooperate and find solutions to mutual problems. One effect this had was the initiations of several separate projects outside of the scope of MBEED.

**Support for current development process.** With experience from earlier company tool introductions, when developers more or less had been forced to adapt to a new tool or work procedure, management found it necessary to practice a new way of change work. One main demand was that the new tool should support the current work procedures of the developers such that they should not be required to adapt in order to use the tool properly. Otherwise there was a risk that they would disapprove of the tool. As a result, the tool was tailored as much as possible according to the expressed needs of the key users. Further, based on a very close cooperation with the tool provider, complete control of the tool is achieved that enables adjustments to organizational needs that might evolve in the future. Therefore, the tool is expected to impose very few restrictions.

One important bonus that the project brought on was that the tool introduction, and especially the discussions about the information model to be adopted, made the work with the development process less abstract and served as a starting point for process development discussions.

**New tool support for MBD.** The tool that was introduced during the project could be described as a light-weight PDM tool for EE development (Figure 4). It is based on a database that is structured according to an information meta-model (information model for short) that was developed in close cooperation with the key users during the project. The tool uses a client-server architecture in which the server holds the information database and supplies an API and a number of general control mechanisms to the clients. These mechanisms include version and variant handling, project phase control and access control to the different model objects. Clients are tailored in a way that each type of user gets its own adjusted view of the complete system model. Hence, a very high degree of tool adaptation was reached in the project, while in the continuation the available clients are expected to be sufficient for most future users.

The tool supports fine grained configuration management, leading to a considerable efficiency improvement in the handling of small amounts of data, one example being the release and review of individual textual requirements. Traceability between modeled objects is supported through the object relations defined in the information model and the fact that clients support browsing between model entities.

The tool also supports automated generation of textual specifications in PDF format once all partial requirements for a given system or component (ECU) have been released and accepted. This enables a problem free integration of the tool in the process where the majority of the developers are still dependant on written documents as input. Further, data specifying graphics such as fonts and icons can be presented in XML documents that can be directly included in the program files for the target ECU.
Figure 4. Architectural overview of the system tool introduced at the studied company.

**Bottom-up approach.** One essential characteristic of the change process was to take on a bottom-up approach in the project. Early involvement and gradual tool development, discussed in the following section, were two important characteristics that permeated the project. The interviewees experienced that the project aim was well expressed among EE developers from the very beginning of the project. It was possible for key users to be engaged in pre-project planning, and this early involvement helped strengthening motivation among developers. A throughout positive attitude among project members also had a positive effect on the outcome of the project in that there was no need to persuade key users to adopt the tool. Therefore one could conclude that the choice of letting key users take an active part resulted in a quick and smooth tool introduction and a firm establishment among users.

In addition, other EE developers in contact with key users expressed a huge interest in taking an active part in the project and using the tool themselves. The introduced system tool was perceived as being easy to learn and intuitive to use and its benefits spread from mouth-to-mouth to other EE developers, causing an organic growth that stretched far beyond the estimated amount of users. In some specific cases, key users acted as teachers introducing new users to the tool. This learning procedure turned out to be a very smooth and quick way to achieve a firm tool introduction. This was a somewhat unexpected effect of the project that was not anticipated from the beginning by project leaders, but in conclusion it can be said that letting the tool grow organically is one way to make the introduction even more time and cost efficient.

In the beginning of the project, a thorough analysis of the key users individual work tasks was made by defining problems in their current work situation, analyzing how their daily work proceeded, and mapping what, when and how information was received and delivered. During the project the tool gradually developed in collaboration with key users. The change work started by consulting key users that had component responsibilities in order to establish the structure of the information needed for component specification. This work was based on close interaction and intensive communication between the system tool supplier and EE developers. The tool development process consisted of iterative loops and fast feedback in that the information model was updated and the tool was adjusted to the user’s needs.
Through detailed knowledge about informal work procedures carried out within EE development, project members obtained a good understanding of perceived and actual problems in their daily work situation. Another effect of the thoroughly made analysis of individual work tasks was that structural problems and possible adjustments to the formal development process were made visible in the organization. Feedback within a week and a short developing period of tailored tool clients kept key users motivated.

**Management support.** Interviewees expressed dissatisfaction with what they felt was weak upper management support for the project. On the contrary, genuine interest was shown by middle management, a circumstance that was expressed by the developers as being an important encouragement. Upper management’s limited interest was expressed by the fact that no long-term objective was set or implemented among the workers. Adequate resources were provided by middle management alone while trying to persuade upper management to invest more time and resources.

Firm management support is needed to prevent people from being insecure about the possibility to allocate time from ordinary work. It is critical that adequate resources are ensured by upper management so that teams can be provided with a supporting attitude. The need has to be anchored throughout the organization and has an expressed system level gain, which was not the case in MBEED. One conclusion is that upper management did not succeed to fulfill these requirements, and that middle management therefore had to take on a bigger responsibility to ensure a successful outcome of the project.

**Parallel work.** With an improved requirement management where requirements are communicated and reviewed individually, immediate changes due to constant access to updated information will be possible due to a steady flow of information. This makes a better work continuity possible reducing the risk of occasional work overload. In turn, this continuous acceptance process enables parallel work, releasing time for developers to take on other work tasks. Using the tool also requires fewer manual work elements due to more automation.

Enhanced possibilities for MBD help improve the communicating ability among developers. One effect of making developers’ work content more explicit to others is a more thorough insight in colleagues work and a better understanding of different needs across engineering disciplines. The information model in a way helps to illustrate the information flow. This results in a deeper understanding among developers of what changes are required and why, so that they can better foresee the expected need of the receiver. In turn, this improves work quality, both when it comes to the product and the individual’s personal work situation.

**Terminology.** One major improvement with the new tool support is that it allows people in the organization to use a language adapted to automotive engineering. The advantages compared to the use of UML were instant. Key users did not have to pass on their work to UML modellers nor did they feel insecure about the proper use of terminology when the risk for misunderstanding external definitions was removed. However, formulating an own language for development led to a need to strictly define the corporate language (which in turn was an unexpected and positive side effect of the project). This was however problematic because engineers from different disciplines use different names for the same thing, and the same name for different things.
**Discussion**

**Roll-out.** The tool introduction in the studied company has so far been limited to minor parts of the EE development. If a roll out will be carried out in the future it will be necessary to take on a broad perspective and consider the effects this change will have on the entire car development. In the following, three circumstances concerning a future roll out are discussed.

First, in the discussed project several end-users and some additional “free riders” were provided with the new tool support, forming small independent islands of users related through loosely connected partial information models. Before considering a broader introduction of the tool it is essential to ascertain that it is possible to merge these different groups of users into one coherent group with one common and agreed upon information model. It is hence necessary to investigate the perceived consequences of allowing scattered groups to evolve on their own, and to consider that this may lead to problems when trying to merge the partial information models. In other words, the integration of different information models may become problematic but a merge is nevertheless necessary in order to structure a complete information model for EE development. The task of defining such a general model still needs to be worked out thoroughly, but through this project part of the initial preparation work has been made.

Second, it is important to consider if, when and how a complete adoption of a new information management tool should be performed. Key users in the studied project are satisfied with the new tool to such a degree that they prefer not to keep on working in the old system. Therefore, in the long run, it is possible to anticipate a wish for a complete tool introduction that includes all EE developers. Today it is possible for key users to work with parallel tools that serve similar purposes but in a longer time perspective this is an unwanted work situation. One reason for this is that it requires a duplication of some work tasks. Further, if a large number of EE developers, but not all, use a specific tool that give them access to mutual information, there still is a need to handle the problems of keeping data consistent with the other tools. Considering that information has to be transferred and stored into the new tool, at some point there is a need to cross over and leave the old information channels behind. However, only when a complete roll out is considered would it be doable to transfer from one information system to another.

Third, it is important to consider possible long-term effects that the tool may have on work tasks, responsibilities and organizational roles. Although the tool is adapted to current work procedures the characteristics of the tool enable more efficient work with a higher degree of automation, which might have an effect on existing work procedures. Reduced work effort and released resources may alter the work situation making it necessary for the company to plan for adjustments in future work procedures.

**Multi-brand adjustments.** Other considerations that need to be taken in this kind of change work concern the risks that external requirements may lead to a major restructuring of the complete development process, reducing the benefits of the so far made efforts. For example there is an ongoing work on aligning the processes between the different companies within the group, which may lead to the subsidiary company having to adapt its development process to that of the parent company. Such a change would necessarily have an effect on the use of development tools in the studied organization. In the project discussed here there are two possible scenarios: either the new tool can be reused to some extent, and adapted to the new process, or the studied organization has to leave the introduced tool behind. In the latter case one outcome of the change work would still be that the structuring of an information model and the
work on defining a common corporate language has started to take shape. It is however vital that
the organization prepares for the risk of not continuing to use the new tool.
Apart from a continued development of the MBEED, there also exist a number of probable
alternatives based on the ongoing standardization efforts mentioned in the introduction. Both the
AUTOSAR (Autosar 2005) and EAST-EEA (EAST-EEA 2005) projects are working on
standards that may come to either replace or at least have impact on the tool and the information
model developed within MBEED. This is of course an important factor to take into account when
considering a continued implementation of the MBEED solution.
One important question related to the above discussion is whether or not it is possible to
introduce the same tool across brands. With a more flexible tool interface that allows individual
tailoring, the system tool introduced for EE development could be made standard for the whole
group of company. This enables continuous adjustments of an emergent information model,
which in turn is an important condition for a successful development of an overarching
information model for EE development. This way, it might be foreseeable to build an
information model from bottom-up that is adjustable to multi-brand development needs.

Conclusions
In conclusion, in this case by taking on a bottom-up approach, keeping a close interaction with
key users while developing the tool and adapting the models according to the terminology used
by the developers, a successful implementation is achieved, and several managerial challenges
are overcome. The fact that the tool was developed gradually in collaboration with the
developers resulted in a tool that supported their current way of working. As a result the
developers expressed a will to assist in the tool development process and to quickly adopt the
new tool. The tool solved immediate problems among EE developers, though, before further
implementation other related activities have to be considered.
By considering product, process and organization, management successfully have laid the
foundation for a well-anchored use of a new tool support within EE development. This provides
means to build the foundation for success in the ongoing work towards model-based
development.

Acknowledgments
The authors would like to thank the participating companies and their employees for taking the
time and effort to share their experiences. The research has been supported by funding by the
Swedish Agency for Innovation Systems, VINNOVA.

References


